Principles of Cryptography Problem Sheet 5

Dr. Kurunandan Jain[1]†,

[1]Department of Mathematics, Amrita Vishwa, Amrita Campus, Kollam-690525, Kerala, India

(Received xx; revised xx; accepted xx)

1) When we studied digital signatures we stated that the sender (or message) authentication always implies data integrity? Why? Is the opposite true too, i.e. does data integrity imply sender authentication. Justify both answers.

2) Design a security service that provides data integrity, data confidentiality and nonrepduation using public key cryptography in two-party communication systems over an insecure channel. Given a rationale that data integrity, confidentiality and nonrepudiation are achieved by your solution. (Consider the corresponding threats in your argumentation)

3) Given an RSA signature scheme with the public key $(n = 9797, e = 131)$, which of the following signatures are valid?

i) $(x = 123, sig(x) = 6292)$

ii) $(x = 4333, sig(x) = 4768)$

iii) $(x = 4333, sig(x) = 1424)$

4) In an RSA digital signature scheme, Bob signs message x_i and sends them together with the signatures s_i and her public key to Alice. Bob's public key is the pair (n, e) her private key is d

Oscar can perform man in the middle attack, i.e. he can replace Bob's public key by his own on the channel. His goal is to alter messages and provide these with a digital signature which will check out correctly on Alice's side. Show everything that Oscar must do for a successful attack.

5) Given an Elgamal signature scheme wit $p = 31$, $\alpha = 3$ and $\beta = 6$. You receive the message $x = 10$ twice with the signatures (r, s):

i) $(17, 5)$

ii) $(13, 15)$

a)Are both signatures valid?

b) How many valid signatures are there for each message x and the specific parameters chosen above?

6) Given the Elgamal signature scheme with the public parameters $p, \alpha \in \mathbb{Z}_{,}^{*}$ and an unknown private key d. Due to faulty implementation, the following dependency between two consecutive ephemeral keys is fulfilled: $K_{E_{i+1}} = K_{E_i} + 1$

Furthermore, two consecutive signatures to the plaintexts x_1 and x_2 are given by (r_1, s_1) and $(r_2, s_2$ are given. Explain how an attacker is able to calculate the private key with the given values.

7) Show how DSA can be attacked if the same ephemeral key is used to sign two different messages

8) We consider three different hash functions which produce output of lengths 64, 128 and 160 bit. After how many random inputs do we have a probability of $\epsilon = 0.5$ for a collision? After how many random input do we have a probability of $\epsilon = 0.1$ for a collision?

† Email address for correspondence: kurunandanj@am.amrita.edu

9) Compute the output of the first round of stage 1 of SHA-1 for a 512-input block of

i) $x = \{0 \cdots 00\}$

ii) $x = \{0 \cdots 01\}$

10) As we have seen MACs can be used to authentic messages. With this problem, we want to show the difference between two protocols one with a MAC, one with a digital signature. In the two protocols, the sending party performs the following operation

Protocol A: $y = e_{k_1}[x||h(k_2||x]$

where x is the message, $h()$ is a hash function such as SHA-1, e is a private-key encryption algorithm, $||$ denotes simple concatenation and k_1, k_2 are secret keys which are only known to send and the receiver.

Protocol B: $y = e_k[x||sig_{k_{pr}}(h(x))]$

Provide a step-by-step description of what the receiver does upon receipt of y. You may wants to draw a block diagram for the process on the receiver's side but that's optional.

11) MACs are, in principles, also vulnerable against collision attacks. We discuss the issue in the following

i) Assume Oscar found a collision between two messages, i.e. $MAC_k(x_1) = MAC_k(x_2)$ show a simple protocol with an attack that is based on a collusion.

ii) Even though the birthday paradox can still be used for constructing collisions, why is it in a practical setting much harder to construct them for MACs then for hash functions? Since this is the case: what security is provided by a MAC with 80-bit output compared to a hash function with 80-bit output?

12) In this exercise, we want to analyse some variants of key derivation. In practice one masterkey K_{MK} is exchanged in a secure way between the involved parties. Afterwards, the session keys are regularly updated by use of key derivation. For this purpose, three different methods are at our disposal:

i) $k_0 = K_{MK}$; $k_{i+1} = k_i + 1$

ii) $k_0 = h(k_{MK})$; $k_{i+1} = h(k_i)$

iii) $k_0 = h(k_{MK})$; $k_{i+1} = h(k_{MK}||i||k_i)$

where $h()$ marks a secure hash function, and k_i is the ith session key.

a) What are the main differences between these methods?

b) Which method provides Perfect Forward Secrecy?

c) Assume Oscar obtains the nth session key (via brute-force). Which sessions can he now decrypt?

d) Which method remains secure if the masterkey k_{MK} is compromised? Give a reason

13) Show that PFS is in fact not given in the simplified Kerberos protocol. Show how Oscar can decrypt past and future communications if:

a) Alice's KEK k_A becomes compromised

b) Bob's KEK k_B becomes compromised.

14) We consider the Diffie-Hellman key exchange protocol. Assume now that Oscar runs an active man-in-the-middle attack against the key exchange as explained in the notes. For the Diffie-Hellman key exchange, use the parameters $p = 467$, $\alpha = 2$, $a = 228$ and $b = 57$ for Alice and Bob, respectively. Oscar uses the value $o = 16$

Compute the key pairs k_{AO} k_{BO} the way Oscar computes them, and the way Alice and Bob compute them